

(05/00)

☒ An ☒ executed ☐ unexecuted declaration of the inventor(s)  
☒ also is enclosed ☐ will follow.

☐ Please amend the specification by inserting before the first line the sentence --This application claims priority under 35 U.S.C. §§119 and/or 365 to \_ filed in \_ on \_; the entire content of which is hereby incorporated by reference.--

☐ A bibliographic data entry sheet is enclosed.

☒ The filing fee has been calculated as follows ☐ and in accordance with the enclosed preliminary amendment:

CLAIMS					
	NO. OF CLAIMS		EXTRA CLAIMS	RATE	FEE
Basic Application Fee					\$690.00 (101)
Total Claims	14	MINUS 20 =		x \$18.00 (103)	
Independent Claims	3	MINUS 3 =		x \$78.00 (102)	
If multiple dependent claims are presented, add \$260.00 (104)					
Total Application Fee					690.00
If verified Statement claiming small entity status is enclosed, subtract 50% of Total Application Fee					345.00
Add Assignment Recording Fee \$40.00 (581) if Assignment document is enclosed					40.00
<b>TOTAL APPLICATION FEE DUE</b>					<b>385.00</b>

☒ This application is being filed without a filing fee. Issuance of a Notice to File Missing Parts of Application is respectfully requested.

☒ A check in the amount of \$ 40.00 is enclosed for the recordation fee due.

☐ Charge \$ \_\_\_\_\_ to Deposit Account No. 02-4800 for the fee due.

☐ The Commissioner is hereby authorized to charge any appropriate fees under 37 C.F.R. §§ 1.16, 1.17 and 1.21 that may be required by this paper, and to credit any overpayment, to Deposit Account No. 02-4800. This paper is submitted in duplicate.

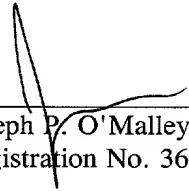
Please address all correspondence concerning the present application to:

James W. Peterson  
Burns, Doane, Swecker & Mathis, L.L.P.  
P.O. Box 1404  
Alexandria, Virginia 22313-1404.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

Date: August 15, 2000

By:   
Joseph P. O'Malley  
Registration No. 36,226

P.O. Box 1404  
Alexandria, Virginia 22313-1404  
(650) 622-2300

005180-005180

Applicant or Patentee: Dirk Slock and Debajyoti Pal  
Application or Patent No.: 09/  
Filed or Issued: August 15, 2000  
For: RELAXED, MORE OPTIMUM TRAINING FOR MODEMS AND THE LIKE

**VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS  
(37 C.F.R. §§ 1.9(f) AND 1.27(c)) - SMALL BUSINESS CONCERN**

I hereby declare that I am

- ☐ the owner of the small business concern identified below:  
☒ an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF CONCERN Excess Bandwidth Corporation  
ADDRESS OF CONCERN 10670 North Tantau Avenue  
Cupertino, CA 95014

I hereby declare that the above-identified small business concern qualifies as a small business concern as defined in 13 C.F.R. § 1.21 for purposes of paying reduced fees under Sections 41(a) and 41(b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average, over the previous fiscal year of the concern, of the persons employed on a full-time, part-time, or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention entitled RELAXED, MORE OPTIMUM TRAINING FOR MODEMS AND THE LIKE, described in

- ☒ the specification filed herewith  
☐ Application No. \_\_\_\_\_, filed \_\_\_\_\_  
☐ Patent No. \_\_\_\_\_, issued \_\_\_\_\_

If the rights held by the above-identified small business concern are not exclusive, each individual, concern, or organization having rights to the invention is listed below,\* and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 C.F.R. § 1.9(c), or by any concern that would not qualify as either a small business concern under 37 C.F.R. § 1.9(d) or a nonprofit organization under 37 C.F.R. § 1.9(e).

\*NOTE: Separate verified statements are required from each named person, concern, or organization having rights to the invention averring to their status as small entities. (37 C.F.R. § 1.27.)

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

☐ individual      ☐ small business concern      ☐ nonprofit organization

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

☐ individual      ☐ small business concern      ☐ nonprofit organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earlier of the issue fee and any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 C.F.R. § 1.28(b).)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code; and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING Debajyoti Pal

TITLE OF PERSON OTHER THAN OWNER Founder, CTO and VP Engineering

ADDRESS OF PERSON SIGNING 10670 North Tantau Avenue

Cupertino, CA 95014

SIGNATURE \_\_\_\_\_ DATE \_\_\_\_\_

"Express Mail" Mailing Label No. EL 521 508 503 US

Date of Deposit August 15, 2000

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above, with sufficient postage affixed, and is addressed to BOX PATENT APPLICATION, Commissioner of Patents, Washington, D.C. 20231.

Wayne Sugihara

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

PATENT  
ATTORNEY'S DOCKET NO. 032478-005

## **RELAXED, MORE OPTIMUM TRAINING FOR MODEMS AND THE LIKE**

### **BACKGROUND OF THE INVENTION**

5

#### **1. Field of the Invention**

The present invention relates to filter adaptation, for example in communications such as wireline communications.

10

#### **2. State of the Art**

15

Broadband communications solutions, such as HDSL2/G.SHDSL (High-Speed Digital Subscriber Line) are increasingly in demand. The ability to achieve high data rates (e.g., 1.5Mbps and above) between customer premises and the telephone system central office over existing (unconditioned) telephone lines requires exacting performance. Various components of a high-speed modem that contribute to this performance require training, e.g., a timing section (PLL, or phase lock loop), an adaptive equalizer, an adaptive echo canceller. Typically, these components are all trained in serial fashion, one after another, during an initial training sequence in which known data is transmitted between one end of the line and the other.

20

25

Equalization is especially critical for HDSL2/G.SHDSL2 modems, which are required to operate over various line lengths and wire models and wirelines with and without bridge taps, with extremely divergent cross-talk scenarios. In general, intersymbol interference (ISI), which equalization aims to eliminate, is the limiting factor in XDSL communications. Hence, good equalization, characterized by the ability to accurately compute the optimal channel equalizer coefficients at the

start-up phase of the modem and adaptively update those coefficients to accommodate any change in the level of cross-talk, is essential to any HDSL2/G.SHDSL system.

Known training methods for high-speed modems suffer from various disadvantages. Existing commercial products invariably use a Least Mean Squares (LMS) training algorithm, which is assumed to converge to an optimal training solution. The LMS algorithm is well-known and has generally been found to be stable and easy to implement. Conventional wisdom holds that the steady-state performance of LMS cannot be improved upon. Despite the widespread use of LMS and its attendant advantages, the adequacy of performance of LMS is being tested by the performance requirements of high-speed modems.

Nor are the alternatives to LMS particularly appealing. Other proposed algorithms have chiefly been of academic interest. The Recursive Least Squares (RLS) algorithm, for example, requires a far shorter training time than LMS (potentially one tenth the training time needed for LMS), but RLS entails exceedingly greater computational complexity. If  $N$  is the total number of taps in an adaptive filter, then the complexity of RLS is roughly  $N^2$ , as compared to  $2N$  for LMS. Also, RLS is less familiar and less tractable, suffering from stability problems.

An improved RLS algorithm ("fast RLS") considerably reduces the computational complexity of RLS, from  $N^2$  to  $28N$ . The original fast RLS algorithm is described in Falconer and Ljung, Application of Fast Kalman Estimation to Adaptive Equalization, *IEEE Transaction on Communications*, Vol. COM-26, No. 10, October 1978, incorporated herein by reference. The fast RLS algorithm, however, requires that training be performed on contiguous data symbols. If training is performed "on-line," then a high-performance processor is required to perform training computations at a rate sufficient to keep pace with the data rate, e.g., 1.5Mbps or greater. Although the computational demand (demand for MIPS) "spikes up" during training, once training is completed, computational demands are

modest. If training is performed "off-line" using stored data samples, then the processor need not keep up with the data rate, reducing peak performance requirements. However, a potentially long sequence of training data must be stored to satisfy the requirement of the algorithm for contiguous data, requiring a sizable memory. Again, the memory requirement, like training itself, is transient. Once training has been completed, the need for such a large memory is removed.

Apart from training, because communications channels vary over time, continuous or periodic filter adaptation is required. In the case of rapidly varying channel conditions, as in wireless communications and especially mobile wireless communications, and in the case of especially long filters relative to adaptation processing power, the use of RLS is indicated. In wireline communications, these conditions are typically not present. Even in the demanding case of HDSL2/G.SHDSL, filter lengths are moderate and channel variation can be considered to be slow. To applicant's knowledge, all wireline modems use LMS "on-line" for non-training filter adaptation.

Although the error criteria used by the LMS and RLS algorithms differ, the prevalent mathematical analysis of these algorithms suggests that the algorithms converge to the same solution, albeit at different rates. LMS uses *mean* squared error, a statistical average, as the error criterion. RLS eliminates such statistical averaging. Instead, RLS uses a deterministic approach based on squared error (note the absence of the word *mean*) as the error criterion. In effect, instead of the statistical averaging of LMS, RLS substitutes temporal averaging, with the result that the filter depends on the number of samples used in the computation. Although the prevalent mathematical analysis predicts equivalent performance for the two algorithms, the mathematical analysis for LMS is approximate only. Although a mathematically exact analysis of LMS has recently been advanced, the overwhelming complexity of that analysis defies any meaningful insight into the behavior of the algorithm and requires numeric solution.



5

## 10

15

20

25

the use of non-contiguous data. This features allows for a wide spectrum of system initialization strategies to be followed, including strategies in which training of multiple subsystems is interleaved to achieve superior training of multiple subsystems and hence the overall system, strategies tailored to meet a specified computational budget, etc.

### BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be further understood from the following description in conjunction with the appended drawings. In the drawings:

- 10 Figure 1 is a block diagram showing portions of a wireline communications transceiver with which the present invention may be used;
- Figure 2 is a block diagram of a decision feedback equalizer of Figure 1;
- 15 Figures 3A-3D are graphs illustrating superior steady-state performance achieved using the RLC-fast algorithm;
- Figure 4 is a chart summarizing the original RLS-fast algorithm and the computational complexity of the algorithm;
- 20 Figure 5 is a chart summarizing the RLC-fast algorithm and the computational complexity of the algorithm;
- Figure 6 is a diagram illustrating conventional training using a single contiguous block of data;
- 25 Figure 7 is a diagram illustrating the inability of conventional adaptation techniques to operate on discontinuous blocks of data;
- 30 Figure 8 is a diagram illustrating the ability, afforded in accordance with one aspect of the present invention, to operate on discontinuous blocks of data;
- Figure 9 is a diagram illustrating further particulars of the reinitialization Figure 8;
- 35 Figure 10 is a flowchart illustrating cooperative, interdependent training of multiple subsystems in accordance with one aspect of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Figure 1, a block diagram is shown illustrating portions of a wireline communications transceiver with which the present invention may be used. The wireline communications transceiver includes a control section, a transmit section, a receive section, and a hybrid section. Within the transceiver, particularly the receive section, may be various subsystems that require training by the control section, for example, a PLL (which may be of digital implementation), an echo canceller and an adaptive equalizer. The training of these subsystems is an interdependent process. For example, some initial training of the PLL may be required prior to any other training. This initial training, however, may not achieve as good results as may be obtained following some training of one or more of the other subsystems, i.e., the echo canceller and adaptive equalizer. As described hereinafter, the present training methods allow for coordinated, interdependent training of multiple sub-systems, offering the potential of substantially improving overall system performance.

Since ISI, which equalization aims to eliminate, is typically the limiting factor in XDSL communications, the focus of the following description will be equalizer training. The same principles, however, may be applied to the training of various different communications subsystems.

Referring to Figure 2, a block diagram is shown of an adaptive decision-adaptive equalizer (DFE) suitable for use in the wireline communications transceiver of Figure 1. An input signal from a communications line (e.g., an HDSL2/G.SHDSL line) is 2x oversampled. The communications line forms the channel for which equalization is to be performed. The samples are input to an adaptive feedforward filter. In conjunction with the filter, a decimation operation is performed. The resulting data stream is applied to a decision element, or "slicer," which produces an output of the equalizer. The output is applied to an adaptive feedback filter, an output of which is summed into the input to the slicer. The DFE structure *per se* is known. Data decisions are filtered by the feedback filter to

eliminate ISI arising from previous pulses. Because the feedback filter compensates for this "past" ISI, the feedforward filter need only compensate for "future" ISI. The equalizer of Figure 2 differs from conventional equalizers in that the filter adaptation is performed using a variant of RLS, *Reinitializable Low Complexity fast*  
5 *least squares (RLC-fast)*, described hereinafter.

An important, even startling, discovery of the present inventors is that RLS-type algorithms, apart from converging faster, converge to a lower noise floor than the LMS algorithm. That is, better equalization can be performed using the RLS-type algorithms than with LMS. This result is illustrated in Figure 3. Only in  
10 the exacting environment of high-speed, wide-band wireline modems such as HDSL2/G.SHDSL does this important difference come to the fore. In fact, experiments have shown that in this environment, even if an adaptive filter is set to a near-optimal solution obtained using an RLS-type algorithm, if the LMS algorithm is then used, the filter settings will actually diverge from the near-optimal solution.  
15 A great incentive therefore exists to use an RLS-type algorithm instead of the prevalent LMS algorithm. Impediments to the use of RLS-type algorithms in this environment include computational complexity and instability.

Although the computational complexity of the fast RLS algorithm is greatly reduced, it remains significant. The computational complexity of adaptation is  
20 measured in terms of the number of multiplications and/or divisions required per filter coefficient times the total number of filter coefficients  $N$  for the structure. Although the present invention may be used with equalizers of other structures and in other applications of adaptive filters, the invention will be described with respect to the exemplary embodiment of Figure 2.

25 Whereas the original fast RLS algorithm requires  $28N$  multiplications and matrix inversion, the computational complexity of the present "RLC-Fast" algorithm is  $22N$  multiplications and involves 2 divisions. This improvement in computational efficiency is achieved by efficiently rewriting the original algorithm. Note that there are algorithms with computational complexity as low as  $17N$ ;

however, they are very susceptible to error accumulation, and are hard to stabilize without the use of additional correction terms. In the case of a fixed-point equalizer implementation, stability is crucial for overall system reliability. The computational complexity of RLC-fast is reduced without significantly degrading the stability of the algorithm.

Referring to Figure 4, a chart summarizing the original fast RLS algorithm is shown.

Referring to Figure 5, a corresponding chart summarizing the RLC-fast algorithm is shown. Figure 5 follows a different but similar notation than that of Figure 4, as set forth in the following table:

**Table 1**

Quantity	Prior Art Algorithm	New Algorithm
Forward predictor coefficients	$A_{Np}(n)$	$A_{fast}$
Backward predictor coefficients	$D_{Np}(n)$	$D_{fast}$
	$E_{pp}(n)$	$E_{fast}$
Forward prediction error	$\varepsilon_p(n)$	$e$
	$\varepsilon_p(n)'$	$e_p$
Backward prediction error	$\eta_p(n)$	$\eta$
	$\mu_p(n)$	$\mu$
	$m_N(n)$	$m$
		$b_n, c_n$ where $\bar{K}_M(n) = [\frac{c_n}{b_n}]$
Newest entering samples	$\xi_p(n)$	$eps_2$
	$S_{MM}, Q_{MM}$ ("rearranging" matrices)	rearrangement performed explicitly

Derivation of the RLC-fast algorithm from the original algorithm and the computational advantages of the RLC-fast algorithm are described in detail in Appendix B.

5 A fixed-point implementation of the RLC-fast algorithm is desirable to reduce computational load and hence increase the speed of the algorithm, as well as to avoid the cost and increased power consumption of a floating-point processor. Because of the underlying stability issues of RLS-type algorithms, such a fixed-point implementation must be carefully considered. The binary point cannot be assumed to be at the beginning just after the sign bit--i.e., all numbers within  $[-1, 1)$ --to  
10 avoid saturation of the variables, since, for some of the internal variables, the actual values may become larger than 1.

Key elements for successful implementation of the RLC-fast algorithm include: (1) Appropriate scaling of the input variables; (2) the position of the binary point for internal variables; (3) efficient internal scaling of the variables after  
15 multiplication and division to reduce loss of precision; (4) complete analysis of the dynamic range of various internal variable; and (5) judicious choice of delta ( $\delta_i$ ) and lambda ( $\lambda$ ) for convergence speed and stability.

A currently preferred implementation assumes 32-bit precision for all the variables, with all the numbers being of signed integer form. The integer numbers  
20 are given a floating point interpretation in which the leading bit is the sign bit, followed by a 5-bit integer part and a 26-bit fractional part. Multiplication and division are performed assuming the foregoing interpretation of the integer variables. There occur two divisions per update. Both are computed as  $1/(1 + x)$  instead of  $1/x$  to reduce the loss in precision.

25 A more detailed description of the RLC-fast algorithm is given in Appendix A (implemented in fixed point arithmetic for the DSP TI-C6x).

Due to the high data rate of the HDSL2/G.SHDSL system, for moderate-size problems ( $N$  about 100), the RLC-fast algorithm, even with its reduced complexity, poses a high computational burden on a typical processor (say, an  $X$  MIPS

processor). In many modems, RLC-fast will be executed only once at the start-up phase of the modem and will not be used in the steady-state, which is the normal operating state for the modem.

5 Hence, although it may be feasible to deploy a high-speed, power-hungry DSP for on-line execution of RLC-fast, such a measure adversely impacts power consumption and may not be cost effective. As a result, off-line implementation of RLC-fast will often be the preferred alternative.

10 However, off-line implementation itself raises problems. The RLS-type algorithm requires a certain data length in order to converge to a near optimal value. The convergence time is a function of the so-called forgetting factor. An aggressive choice of the forgetting factor can be used to reduce the required data length but at the cost of stability.

15 A reasonable choice for the forgetting factor may require a long data length (say, 100N) for convergence. This in turn implies a large storage requirement even for a moderate size problem. Once again, if this memory is only used during the start-up phase, a straight-forward implementation wastes large amounts of silicon and results in inefficient design.

20 The original fast RLS algorithm offers no solution to the foregoing problem. Referring more particularly to Figure 6, the original fast RLS algorithm requires the input data stream to be contiguous. If there is a break in the input data stream, the only way to use the new data in the original approach is to restart the algorithm all over again as illustrated in Figure 7. Of course, the algorithm can be trained with smaller size blocks of data, but only at the cost of reduced performance. That is, the advantage of Figure 3 would be sacrificed.

25 To circumvent the requirement of a contiguous data stream, RLC-fast uses a re-initialization scheme that allows the use of a non-contiguous data block without restarting the algorithm. At start-up, the algorithm is initialized in the usual way. However, the algorithm can be stopped at any time and started at a later time with a new initialization. This manner of operations is illustrated in Figure 8. No

difference in performance is observed if individual data blocks are not too small (say, no smaller than  $10N$ ). Hence, storage requirements may be reduced by an order of magnitude (e.g.,  $10N$  instead of  $100N$ ).

5       The particulars of re-initialization are illustrated in Figure 9. Instead of setting the intermediate variables to zero or a scaled identity matrix, the previous values are used for all variables except  $X_{fast}$ . The variables  $A_{fast}$ ,  $F_{fast}$ ,  $K_{fast}$ ,  $b_n$ ,  $D_{fast}$  and  $C_{fast}$  are all stored for this purpose.

10       The foregoing re-initialization capability allows for a store/process mode of operation. More particularly, even with the reduced complexity of RLC-fast, the amount of computation required for *real-time* processing of moderate size problems can be prohibitive for most DSPs due to the high data rate of the system. To alleviate this problem, a store/process mode of operation is followed in which, during the first half of a cycle, a small block of data (e.g., size  $10N$ ) is stored, and during the second half of the cycle, the data is processed to update the filter  
15       coefficients. Instead of operating in real-time, since the data is stored, each update need not be finished within the sample time  $T$ . Instead, the computation can be distributed over multiple sample periods.

20       One approach is to partition the computation of the update for each data sample in small enough segments such that an individual segment can be finished in one sample time. The smaller the partition, the less processing is required each sample period. Total time to finish the update increases. Hence, store/process operation, along with partitioning of the update computation, provides a flexible mechanism that allows for trade-off between processing load and total time to process a data block. Without the capability of re-initialization, this flexibility is not  
25       obtainable.

      The same flexibility may be extended from the adaptive equalizer or other isolated sub-system to the system as a whole, in such as way as to achieve not only great flexibility but also improved performance. In reality, the performance of each sub-system is interdependent on the performance of other sub-systems and should



not be viewed in isolation. Referring to Figure 10, for example, the performance of the clock recovery circuit of the PLL block is influenced by the performance of the echo canceller and vice versa. The same is true for the performance of the echo canceller and the equalizer. The better the echo cancellation is, the better the  
5 equalizer performance will be. As a consequence, there will be fewer errors in the decision. These more accurate decisions can be used to improve the echo cancellation performance. This improved echo cancellation can be utilized to reduce the jitter performance of the PLL.

10 It will be appreciated by those of ordinary skill in the art that the invention can be embodied in other specific forms without departing from the spirit or essential character thereof. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than the foregoing description, and all changes which come within the meaning and range of equivalents thereof are  
15 intended to be embraced therein.

## APPENDIX A

### A Detailed Description of EBC-Fastalgo

#### Step 1 Initialization:

$$\begin{aligned}A_{fast} &= 0, \quad (N \times 3) \\D_{fast} &= 0, \quad (N \times 3) \\F_{fast} &= \delta_i I, \quad (3 \times 3) \\K_{fast} &= 0, \quad (N \times 1) \\X_{fast} &= 0, \quad (N \times 1) \\C_{fast} &= 0, \quad (N \times 1) \\decision &= 0. \quad (1 \times 1).\end{aligned}$$

#### Step 2 Data Acquisition:

$$\begin{aligned}p_3 &= [X_{fast}(FFE_{LEN} - 2) \ X_{fast}(FFE_{LEN} - 1) \ X_{fast}(N - 1)], \\eps_2 &= [y(2); y(1); decision]; \\decision &= txsignal.\end{aligned}$$

#### Step 3 Update Computation:

$$e = eps_2 + A_{fast}^T X_{fast}.$$

```
e[0]=eps2[0];
e[1]=eps2[1];
e[2]=eps2[2];
for(ind=0;ind<N;ind++)
{
e[0]+= mpy n(A fast[ind],X fast[ind]);
e[1]+= mpy n(A fast[ind+N],X fast[ind]);
e[2]+= mpy n(A fast[ind+N 2],X fast[ind]);
}
```

$$A_{fast} = A_{fast} - K_{fast} e^T.$$

```
for(ind=0;ind<N;ind++)
{
ind2=ind+N;
ind3=ind+N 2;
```

## APPENDIX A (continued)

```
A fast[ind] -= mpy n(K fast[ind],e[0]);
A fast[ind2] -= mpy n(K fast[ind],e[1]);
A fast[ind3] -= mpy n(K fast[ind],e[2]);
}
```

$$e_p = eps2 + A_{fast}^T X_{fast}.$$

Equivalently,

$$e_p = e(1 - K_{fast}^T X_{fast}).$$

```
temp fast=inp(K fast,X fast,N);
temp fast=FIXED ONE-temp fast;
ep[0]=mpy n(temp fast,e[0]);
ep[1]=mpy n(temp fast,e[1]);
ep[2]=mpy n(temp fast,e[2]);
```

$$F_{fast} = \lambda_i F_{fast}.$$

```
mat scal(lambda,F fast,MM,F fast);
```

$$\begin{aligned} F_{fast} &= F_{fast} - (F_{fast} e_p e^T F_{fast}) \frac{1}{(1 + e^T F_{fast} e_p)}, \\ t_3 &= F_{fast} e_p, \end{aligned}$$

We use the following equivalent computation:

$$\begin{aligned} t_3 &= F_{fast} \frac{e_p}{1 + e^T F_{fast} e_p}, \\ F_{fast} &= F_{fast} - t_3 e^T F_{fast}. \end{aligned}$$

```
mat cvec(F fast,ep,M,temp vecM);
temp fast=inp(e,temp vecM,M);
temp fast=div wor(temp fast+FIXED ONE);
mat scal(temp fast,temp vecM,M,temp vecM);
mat rvec(F fast,e,M,temp vecM1);
outp(temp vecM,temp vecM1,M,temp matM);
for(ind=0;ind<MM;ind++)
```

## APPENDIX A (continued)

```
{
F fast[ind]=F fast[ind]-temp matM[ind];
}
```

$$temp_{vec} = F_{fast} e_p.$$

```
mat cvec(F fast,ep,M,temp vecM);
```

$$b_n = K_{fast} + A_{fast} * t_3;$$

```
for(ind=0;ind<N;ind++)
{
bn[ind]=K fast[ind];
bn[ind]+=mpy n(A fast[ind],temp vecM[0]);
bn[ind]+=mpy n(A fast[ind+N],temp vecM[1]);
bn[ind]+=mpy n(A fast[ind+N 2],temp vecM[2]);
}
```

$$m = [t_3(1:2); b_n(1:FFE_{LEN} - 2); t_3(3); b_n(FFE_{LEN} + 1: N - 1)].$$

```
m vec[0]=temp vecM[0];
m vec[1]=temp vecM[1];
for(ind=2;ind<FFE LEN;ind++)
{
m vec[ind]=bn[ind-2];
}
m vec[FFE LEN]=temp vecM[2];
for(ind=FFE LEN+1;ind<N;ind++)
{
m vec[ind]=bn[ind-1];
}
```

$$\mu = [b_n(FFE_{LEN} - 1: FFE_{LEN}); b_n(N)].$$

```
mu[0]=bn[FFE LEN-2];
mu[1]=bn[FFE LEN-1];
mu[2]=bn[N-1];
```

## APPENDIX A (continued)

$$X_{fast} = [eps_2(1:2); X_{fast}(1:FFE_{LEN}-2); eps_2(3); X_{fast}(FFE_{LEN}+1:N-1)].$$

```

for(ind=FFE LEN-1;ind>1;ind--)
{
X fast[ind]=X fast[ind-2];
}
X fast[1]=eps2[1];
X fast[0]=eps2[0];

for(ind=FBE LEN-1;ind>0;ind--)
{
X fast[ind+FFE LEN]=X fast[FFE LEN+ind-1];
}
X fast[FFE LEN]=eps2[2];

```

$$\eta = p_3 + D_{fast}^T X_{fast}.$$

```
eta[0]=p3[0];
eta[1]=p3[1];
eta[2]=p3[2];

for(ind=0;ind<N;ind++)
{
eta[0]+=mpy n(D fast[ind],X fast[ind]);
eta[1]+=mpy n(D fast[ind+N],X fast[ind]);
eta[2]+=mpy n(D fast[ind+N 2],X fast[ind]);
}
```

$$\begin{aligned} D_{fast} &= D_{fast} - m\eta^T(\mathbf{I} + \mu\eta^T/(1 - \eta^T\mu)), \\ K_{fast} &= m - D_{fast}\mu. \end{aligned}$$

We use the equivalent computation:

$$\begin{aligned} K_{fast} &= (m - (D_{fast}\mu))/(1 - \eta^T\mu), \\ D_{fast} &= D_{fast} - K_{fast}\eta^T. \end{aligned}$$

APPENDIX A (continued)

```
temp fast=inp(eta,mu,M);
temp fast=div wor(FIXED ONE-temp fast);
for(ind=0;ind<N;ind++)
{
ind2=ind+N;
ind3=ind+N 2;
K fast[ind]=m vec[ind];
K fast[ind]-=mpy n(D fast[ind],mu[0]);
K fast[ind]-=mpy n(D fast[ind2],mu[1]);
K fast[ind]-=mpy n(D fast[ind3],mu[2]);
K fast[ind]=mpy n(K fast[ind],temp fast);
}
```

```
for(ind=0;ind<N;ind++)
{
ind2=ind+N;
ind3=ind+N 2;
D fast[ind] -=mpy n(K fast[ind],eta[0]);
D fast[ind2] -=mpy n(K fast[ind],eta[1]);
D fast[ind3] -=mpy n(K fast[ind],eta[2]);
}
```

$$x_{est} = C_{fast} X_{fast}.$$

```
xest=inp(X fast,C fast,N);
```

$$error = decision - x_{est}.$$

```
error xest= decision-xest;
```

$$C_{fast} = C_{fast} + error * K_{fast}.$$

```
if(((gdat.fastbuffoffset)*2)>FFE LEN)
{
for(ind=0;ind<N;ind++)
{ C fast[ind] +=mpy n(K fast[ind],error xest);
}
}
```

## APPENDIX B

### Step-by-step derivation of the RLC-fast algorithm from fast algorithm

#### Step 1 Initialization:

$$\begin{aligned} A_{fast} &= 0, \quad (N \times 3) \\ D_{fast} &= 0, \quad (N \times 3) \\ F_{fast} &= \delta_i I, \quad (3 \times 3) \\ K_{fast} &= 0, \quad (N \times 1) \\ X_{fast} &= 0, \quad (N \times 1) \\ C_{fast} &= 0, \quad (N \times 1) \\ decision &= 0. \quad (1 \times 1). \end{aligned}$$

The only difference is in the definition of  $F_{fast}$ . Instead of defining the variable  $E_{fast}$ , we define its inverse, i.e.,  $F_{fast} = E_{fast}^{-1}$ . Hence,  $\delta_i = 1/\delta$ .

#### Step 2 Data Acquisition:

$$\begin{aligned} p_3 &= [X_{fast}(FFE_{LEN} - 2) \ X_{fast}(FFE_{LEN} - 1) \ X_{fast}(N - 1)], \\ eps_2 &= [y(2); y(1); decision]; \\ decision &= txsignal. \end{aligned}$$

This phase is identical to the data acquisition phase of fast algorithm.

#### Step 3 Update Computation:

The first two equations are identical:

$$\begin{aligned} e &= eps_2 + A_{fast}^T X_{fast}, \\ A_{fast} &= A_{fast} - K_{fast} e^T. \end{aligned}$$

Fast algorithm:

$$e_p = eps_2 + A_{fast}^T X_{fast}.$$

Using the last relation for  $A_{fast}$ , we write

$$\begin{aligned} A_{fast}^T X_{fast} &= (A_{fast} - K_{fast} e^T)^T X_{fast}, \\ &= A_{fast}^T X_{fast} - e K_{fast}^T X_{fast}, \\ &= e - eps_2 - e K_{fast}^T X_{fast}, \end{aligned}$$

## APPENDIX B (continued)

where in the last equation we used the definition of  $e$ . Substituting this expression for  $A_{fast}^T X_{fast}$  back to the equation for  $e_p$ , we get an equivalent express for  $e_p$  as

$$e_p = e(1 - K_{fast}^T X_{fast}).$$

Motivation: Reduction in computation from  $mN$  to  $N + \tilde{m}$ .

Fast algorithm:

$$\begin{aligned} E_{fast} &= \lambda E_{fast} + e_p e^T, \\ b_n &= K_{fast} + A_{fast} E_{fast}^{-1} e_p, \\ c_n &= E_{fast}^{-1} e_p, \end{aligned}$$

First notice that,

$$\begin{aligned} b_n &= K_{fast} + A_{fast} E_{fast}^{-1} e_p, \\ &= K_{fast} + A_{fast} c_n. \end{aligned}$$

Now,

$$\begin{aligned} c_n &= E_{fast}^{-1} e_p, \\ &= (\lambda E_{fast} + e_p e^T)^{-1} e_p, \\ &= \lambda_i E_{fast}^{-1} \left[ 1 - \frac{e_p e^T \lambda_i E_{fast}^{-1} e_p}{1 + e^T \lambda_i E_{fast}^{-1} e_p} \right], \end{aligned}$$

where in the last line we used the matrix inversion lemma and  $\lambda_i = 1/\lambda$ . Substituting  $F_{fast} = E_{fast}^{-1}$  and rearranging, we get

$$c_n = \lambda_i F_{fast} \frac{e_p}{1 + e^T \lambda_i F_{fast} e_p}.$$

After resequencing, we obtain

$$\begin{aligned} F_{fast} &= \lambda_i F_{fast}, \\ c_n &= F_{fast} \frac{e_p}{1 + e^T F_{fast} e_p}, \\ F_{fast} &= F_{fast} - c_n e^T F_{fast}, \\ b_n &= K_{fast} + A_{fast} c_n. \end{aligned}$$

Motivation: Reduction in computation from  $mN + 4m^2$  multiplications and an  $(m \times m)$  matrix inversion to  $mN + 4m^2 + 2m$  multiplication and a scalar division. This also improves the stability of the algorithm.



## APPENDIX B (continued)

Next, four equations are identical:

$$\begin{aligned} m &= [c_n(1:2); b_n(1:FFE_{LEN}-2); c_n(3); b_n(FFE_{LEN}+1:N-1)], \\ \mu &= [b_n(FFE_{LEN}-1:FFE_{LEN}); b_n(N)], \\ X_{fast} &= [eps_2(1:2); X_{fast}(1:FFE_{LEN}-2); eps_2(3); X_{fast}(FFE_{LEN}+1:N-1)], \\ \eta &= p_3 + D_{fast}^T X_{fast}. \end{aligned}$$

Fast Algorithm:

$$\begin{aligned} D_{fast} &= (D_{fast} - m\eta^T)(\mathbf{I} - \mu\eta^T)^{-1}, \\ K_{fast} &= m - D_{fast}\mu. \end{aligned}$$

Substituting the first relation for  $D_{fast}$  into the second equation and using matrix inversion lemma, we get

$$\begin{aligned} K_{fast} &= m - (D_{fast} - m\eta^T)(\mathbf{I} - \mu\eta^T)^{-1}\mu, \\ &= m - (D_{fast} - m\eta^T) \left( \mathbf{I} + \frac{\mu\eta^T}{1 - \eta^T\mu} \right) \mu, \\ &= m - (D_{fast} - m\eta^T)\mu \left( 1 + \frac{\eta^T\mu}{1 - \eta^T\mu} \right), \\ &= m - \frac{(D_{fast} - m\eta^T)\mu}{1 - \eta^T\mu}, \\ &= \frac{m - (D_{fast}\mu)}{1 - \eta^T\mu}. \end{aligned}$$

Now,

$$\begin{aligned} D_{fast} &= (D_{fast} - m\eta^T)(\mathbf{I} - \mu\eta^T)^{-1}, \\ &= (D_{fast} - m\eta^T) \left( \mathbf{I} + \frac{\mu\eta^T}{1 - \eta^T\mu} \right), \\ &= D_{fast} \left( \mathbf{I} + \frac{\mu\eta^T}{1 - \eta^T\mu} \right) - m\eta^T \left( \mathbf{I} + \frac{\mu\eta^T}{1 - \eta^T\mu} \right), \\ &= D_{fast} + \frac{D_{fast}\mu\eta^T}{1 - \eta^T\mu} - \left( m\eta^T + \frac{m\eta^T\mu\eta^T}{1 - \eta^T\mu} \right), \\ &= D_{fast} + \frac{D_{fast}\mu\eta^T}{1 - \eta^T\mu} - \frac{m\eta^T}{1 - \eta^T\mu}, \\ &= D_{fast} + \frac{D_{fast}\mu\eta^T - m\eta^T}{1 - \eta^T\mu}, \\ &= D_{fast} - \frac{(m - D_{fast}\mu)}{1 - \eta^T\mu} \eta^T, \\ &= D_{fast} - K_{fast}\eta^T, \end{aligned}$$

APPENDIX B (continued)

where in the last step we substituted the express for  $K_{fast}$ . Hence,

$$\begin{aligned} K_{fast} &= (m - (D_{fast}\mu))/(1 - \eta^T\mu), \\ D_{fast} &= D_{fast} - K_{fast}\eta^T. \end{aligned}$$

Motivation: Reduction in computation from  $(m+2)mN+m^2$  multiplications and an  $(m \times m)$  matrix inversion to  $(2m+1)N+m$  multiplications and a scaler division.

The last three equation are identical:

$$\begin{aligned} x_{est} &= C_{fast}X_{fast}, \\ error &= decision - x_{est}, \\ C_{fast} &= C_{fast} + errorK_{fast}. \end{aligned}$$

What is claimed is:

1. A high-speed, broadband, wireline modem including an adaptive equalizer having both a training mode and a decision-directed, non-training mode, the adaptive equalizer comprising:

5 at least one of: a forward path coupled to receive the signal samples, the forward path including a forward filter and a decision element, and a feedback path coupled between an output of the decision element and an input of the decision element, the feedback path including a feedback filter; and

10 means for adapting the one of said forward filter and said feedback filter based on a least squares error criterion, as distinguished from a least mean squares error criterion.

15 2. The apparatus of Claim 1, further comprising a memory for storing received signal samples.

3. The apparatus of Claim 1, comprising both said feedforward path and said feedback path.

20 4. The method of Claim 1, wherein the means for adapting operates during decision-directed mode.

25 5. The method of Claim 1, wherein the combined length of the forward filter and the feedback filter is moderate relative to adaptation processing power.

6. The method of Claim 1, wherein adaptation is performed using fixed-point arithmetic.

7. The method of Claim 1, wherein said means for adapting performs substantially the following computation:

$$e_p = e(1 - K^{T_{fast}} X_{fast}).$$

8. The method of Claim 1, wherein said means for adapting performs substantially the following computations:

$$F_{fast} = \lambda_t F_{fast},$$

$$c_n = F_{fast} \frac{e_p}{1 + e^{T_{fast}} F_{fast} e_p},$$

$$F_{fast} = F_{fast} - c_n e^{T_{fast}} F_{fast},$$

$$b_n = K_{fast} + A_{fast} c_n.$$

9. The method of Claim 1, wherein said means for adapting performs substantially the following computations:

$$K_{fast} = (m - (D_{fast}\mu))/(1 - \eta^T\mu),$$

$$D_{fast} = D_{fast} - K_{fast}\eta^T.$$

10. The method of Claim 1, wherein a routine for updating said one of said forward filter and said feedback filter performs no more than 22N multiplies, where N is the number of filter taps, and wherein no distinct stabilization quantity is computed.

11. A method of performing adaptation of an adaptive filter using an LS-type adaptation algorithm, comprising:  
storing a first block of data;  
initializing the adaptation algorithm;  
processing the first block of data using the adaptation algorithm;  
storing a second block of data; and  
re-initializing the adaptation algorithm using results from processing the first block of data.

12. The method of Claim 11, wherein the adaptive filter is used in a communication system in which symbols are communicated, one symbol per symbol period, further comprising:

starting and finishing processing of a block of data within a portion of a single symbol period, a remaining portion of the symbol period being used for other computations.

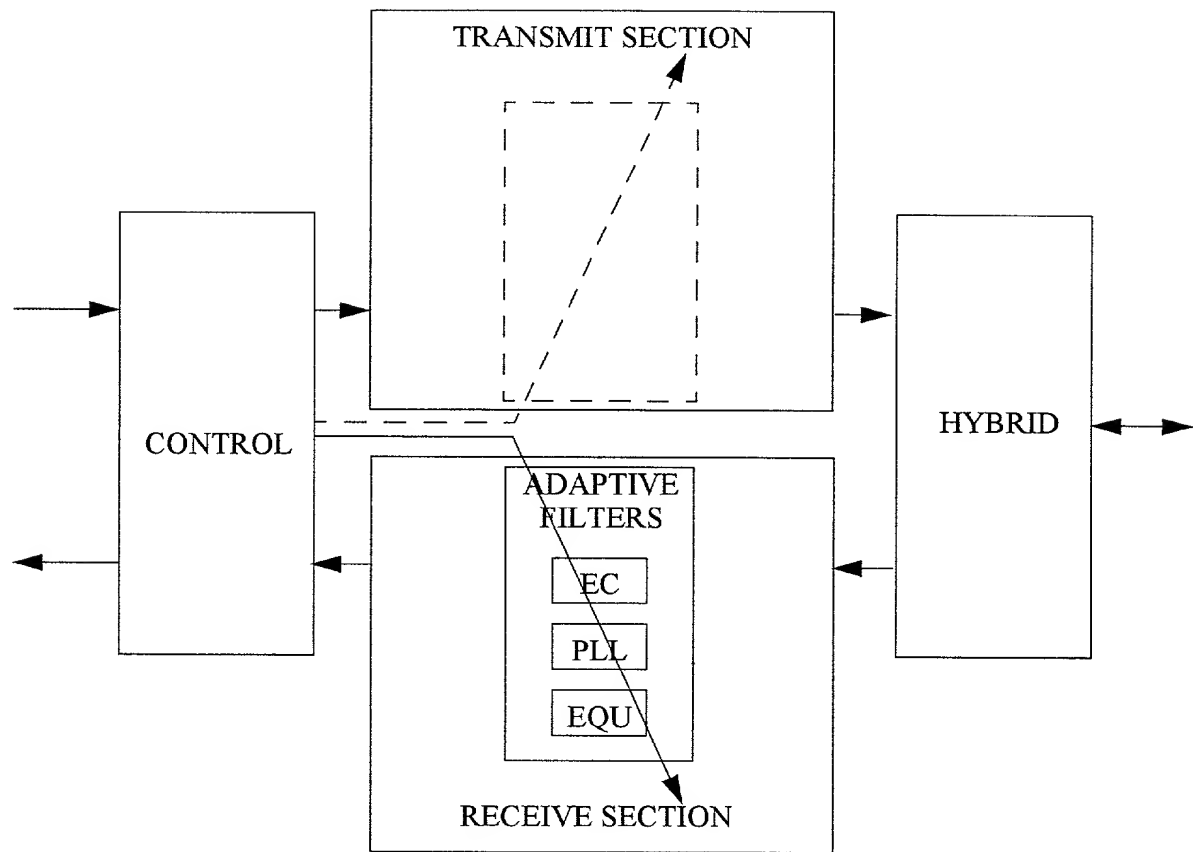
13. The method of Claim 12, further comprising selectably fixing said remaining portion by fixing a size for the data blocks.

- 5 14. A method of modem training in which the modem includes multiple subsystems provided with respective adaptive filters, comprising:
- partially training the adaptive filter of a first subsystem by applying a first training algorithm to first data to obtain first filter settings;
- at least partially training the adaptive filter of a second subsystem;
- and
- 10 completing training of the adaptive filter of the first subsystem by using for initialization at least some of said first filter settings and applying the first training algorithm to second data not contiguous with said first data in a received data stream.

ABSTRACT

A high-speed, broadband wireline modem using the least squares error criterion rather than a least mean squares criterion is described.

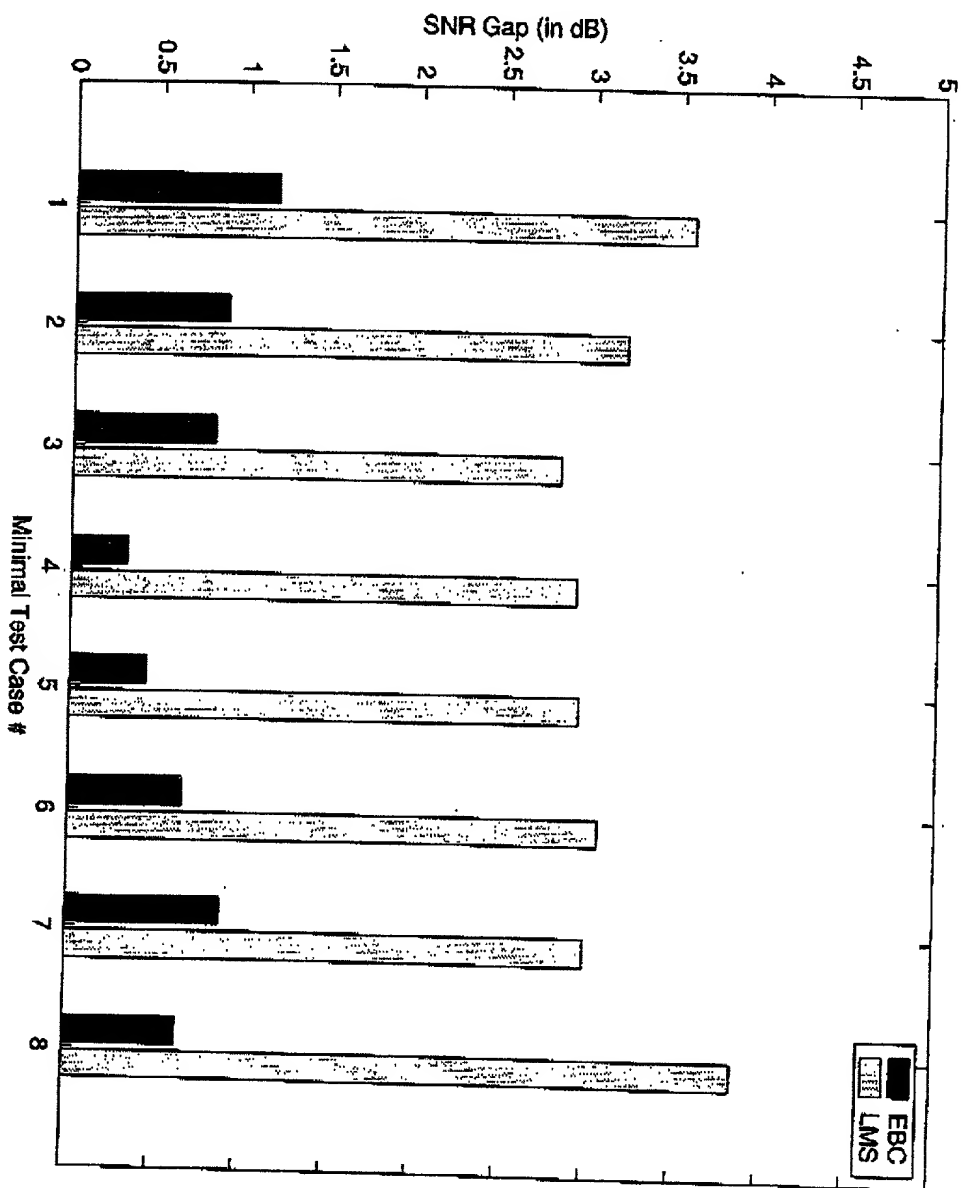
5



*Fig. 1*







3a  
Figure 1: Performance Gap for the HDL2 Minimal Test Set.



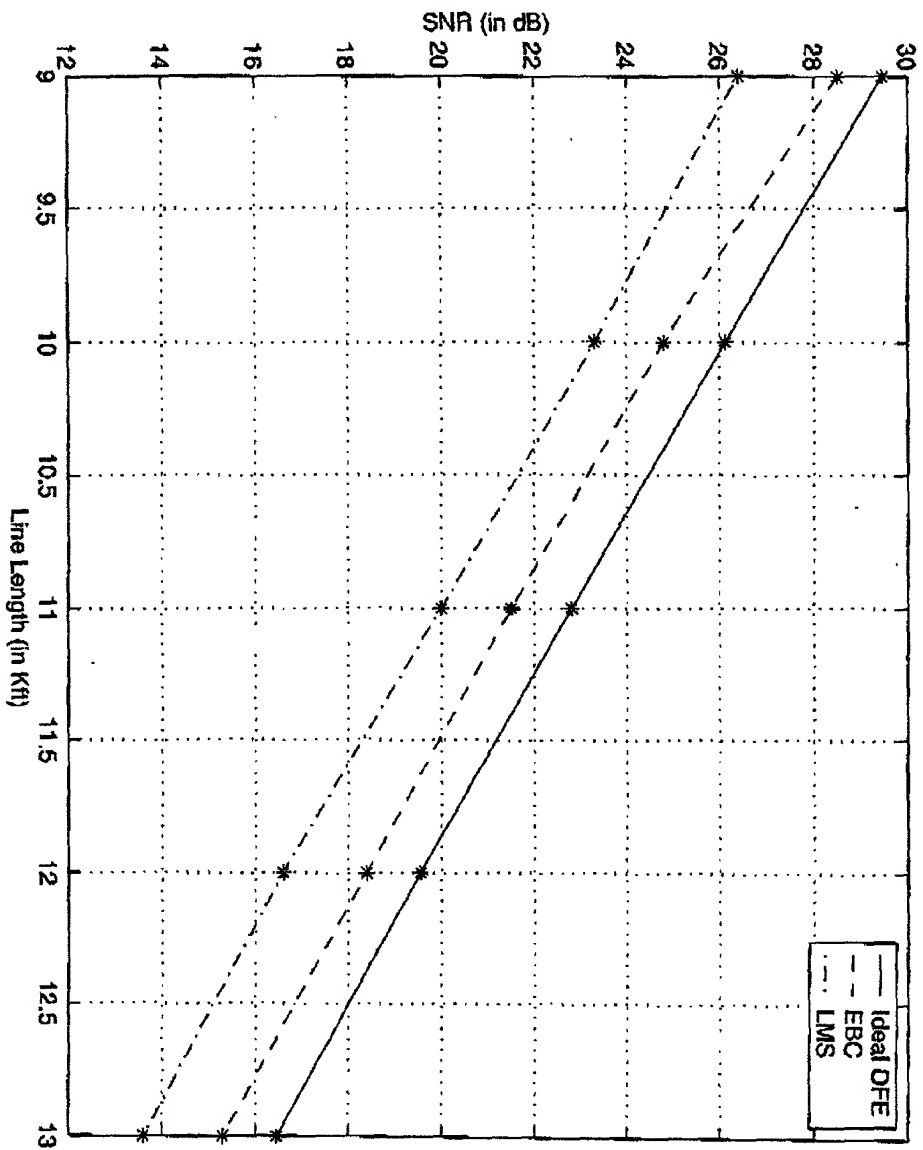


Figure 3c: Line Length vs SNR for the Test Case #6.

09030515.081500

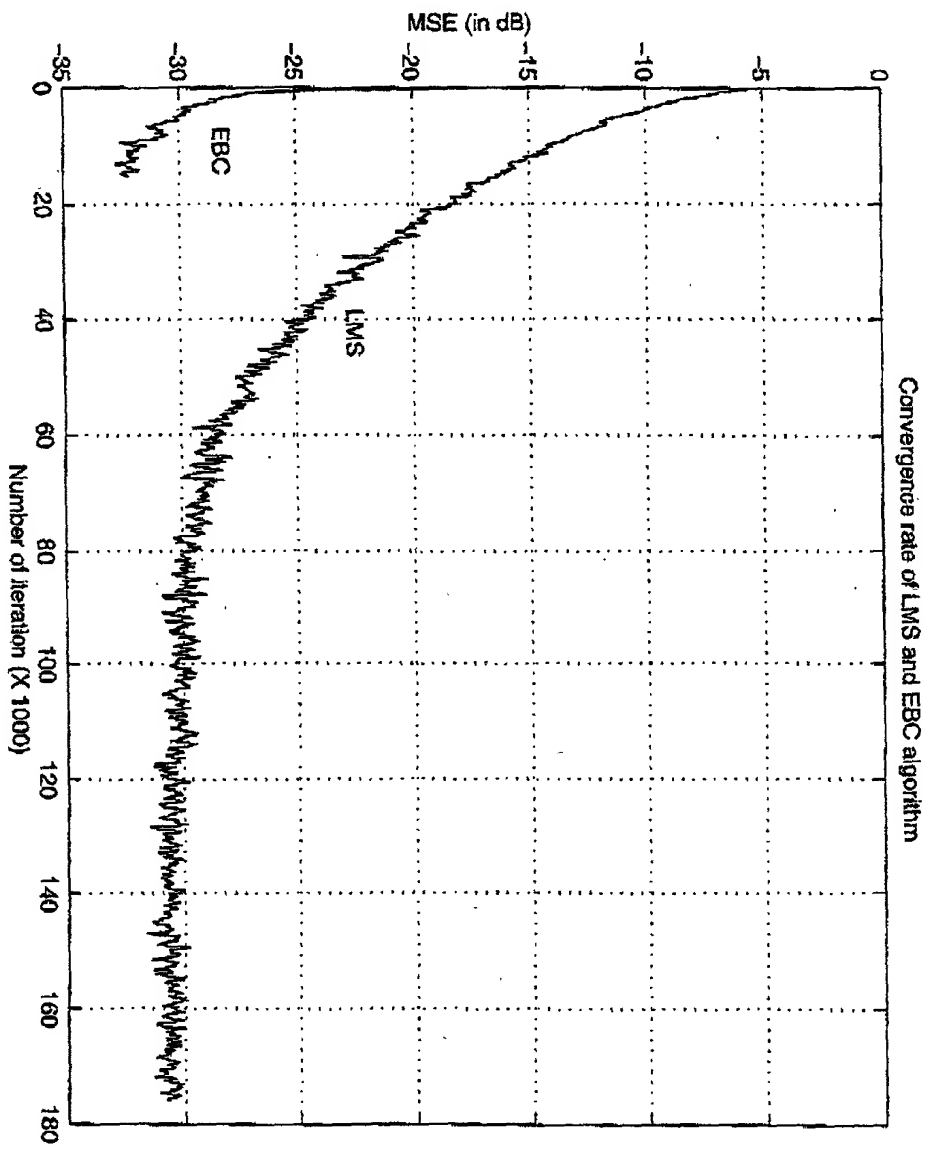


Figure 8: <sup>2d</sup> Comparison of convergence of LMS and EBC algorithm

09639515, 081500

Step	Update Equation	# of Mul
1	$e \leftarrow eps_2 + A_f^T X_f$	$mN$
2	$A_f \leftarrow A_f - K_f e e^T$	$mN$
3	$e_p \leftarrow eps_2 + A_f^T X_f$	$mN$
4	$E_f \leftarrow \lambda E_f + e_p e_p^T$	$2m^2$
5	$b_n \leftarrow K_f + A_f E_f^{-1} e_p$	$m^3 + m^2 + mN$
6	$c_n \leftarrow E_f^{-1} e_p$	$m^2$
7	$m \leftarrow [c_n(1:2); b_n(1:N_{ff}-2); c_n(3); b_n(N_{ff}+1:N-1)]$	0
8	$\mu \leftarrow [b_n(N_{ff}-1:N_{ff}); b_n(N)]$	0
9	$X_f \leftarrow [eps_2(1:2); X_f(1:N_{ff}-2); eps_2(3); X_f(N_{ff}+1:N-1)]$	0
10	$\eta \leftarrow p_3 + D_f^T X_f$	$mN$
11	$D_f \leftarrow (D_f - m\eta^T)(I - \mu\eta^T)^{-1}$	$m^2(N+1) + m^3 + mN$
12	$K_f \leftarrow m - D_f^T \mu$	$mN$
13	$x_{est} \leftarrow C_f X_f$	$N$
14	$error \leftarrow decision - x_{est}$	0
15	$C_f \leftarrow C_f + error K_f$	$N$
Total complexity		$(m^2 + 7m + 2)N + 5m^2 + 2m^3$

Table 1: Update equations and complexity of the original RLS fast algorithm.

F84

Step	Update Equation	# of Mul
1	$e \leftarrow eps_2 + A_f^T X_f$	$mN$
2	$A_f \leftarrow A_f - K_f e^T$	$mN$
3	$e_p \leftarrow e(1 - K_f^T X_f)$	$N + m$
4	$F_f \leftarrow \lambda F_f$	$m^2$
5	$c_n \leftarrow F_f \frac{e_p^T}{1 + e^T F_f e_p}$	$2m(m + 1)$
6	$F_f \leftarrow F_f - c_n e^T F_f$	$m(m + 1)$
7	$b_n \leftarrow K_f + A_f c_n$	$mN$
8	$m \leftarrow [c_n(1 : 2); b_n(1 : N_{ff} - 2); t_3(3); b_n(N_{ff} + 1 : N - 1)]$	0
9	$\mu \leftarrow [b_n(N_{ff} - 1 : N_{ff}); b_n(N)]$	0
10	$X_f \leftarrow [eps_2(1 : 2); X_f(1 : N_{ff} - 2); eps_2(3); X_f(N_{ff} + 1 : N - 1)]$	0
11	$\eta \leftarrow p_3 + D_f^T X_f$	$mN$
12	$K_f \leftarrow (m - (D_f \mu)) / (1 - \eta^T \mu)$	$(m + 1)N + m$
13	$D_f \leftarrow D_f - K_f \eta^T$	$mN$
14	$x_{est} \leftarrow C_f X_f$	$N$
15	$error \leftarrow decision - x_{est}$	0
16	$C_f \leftarrow C_f + error K_f$	$N$
	Total complexity	$(6m + 4)N + 4m^2 + 5m$

Fig 5

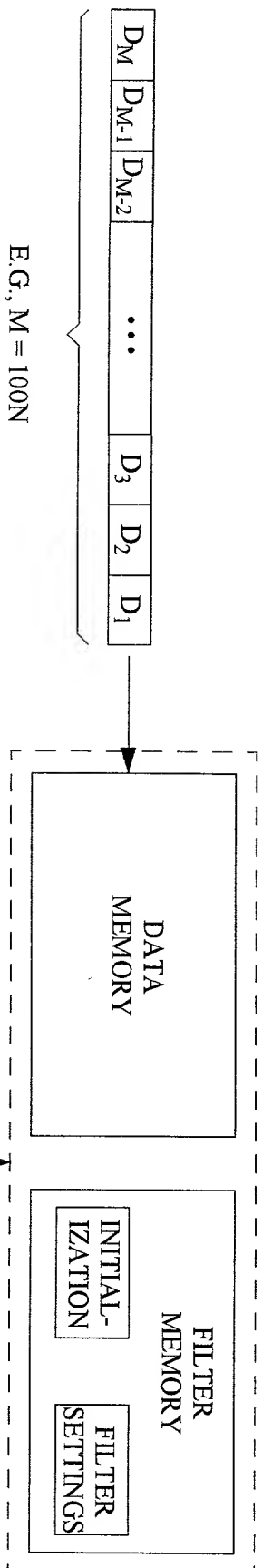


Fig. 6  
(Prior Art)

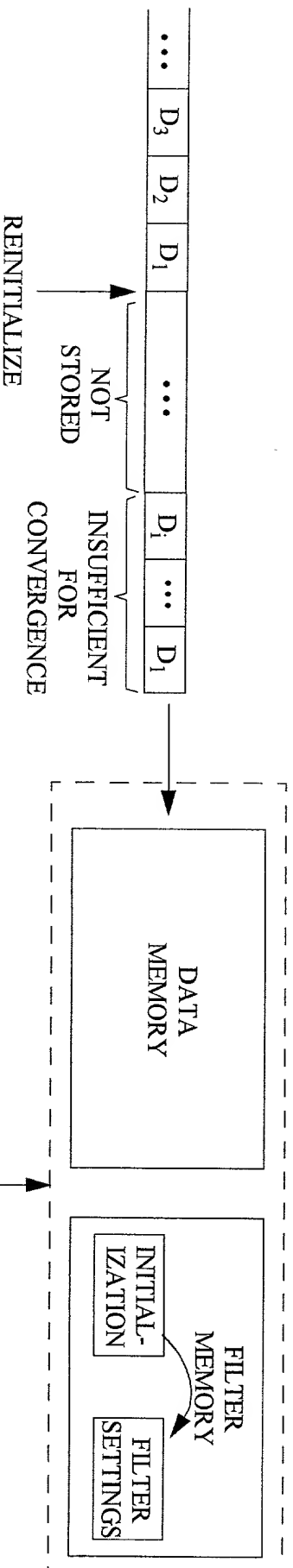


Fig. 7  
(Prior Art)



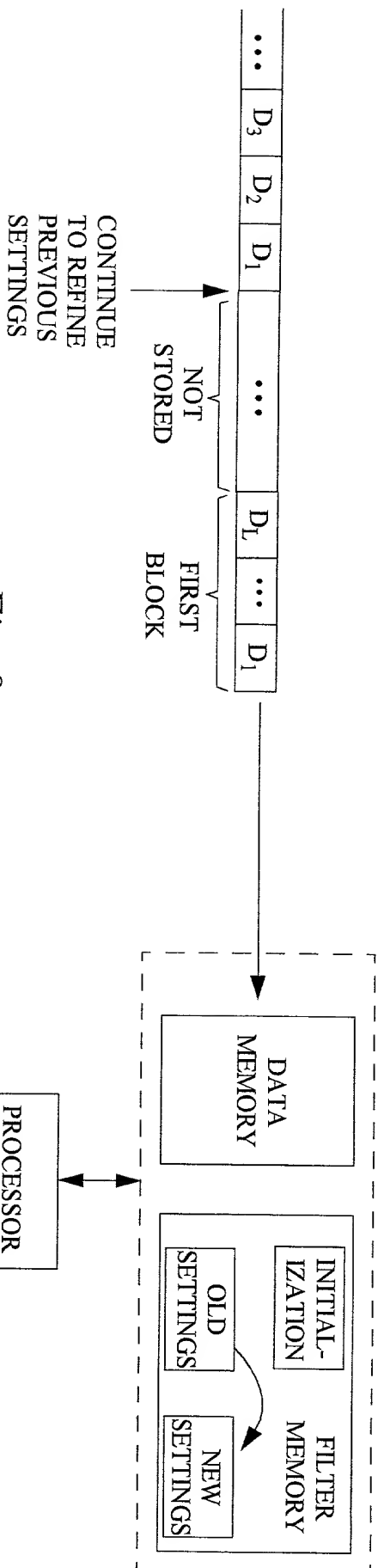


Fig. 8

$A_{fast} \leftarrow$  stored  $A_{fast}$ ,  $(N \times 3)$   
 $D_{fast} \leftarrow$  stored  $D_{fast}$ ,  $(N \times 3)$   
 $F_{fast} \leftarrow$  stored  $F_{fast}$ ,  $(3 \times 3)$   
 $K_{fast} \leftarrow$  stored  $K_{fast}$ ,  $(N \times 1)$   
 $C_{fast} \leftarrow$  stored  $C_{fast}$ ,  $(N \times 1)$   
 $X_{fast} \leftarrow$  fill with new data in order,  $(N \times 1)$   
 $decision \leftarrow$  most current input data,  $(1 \times 1)$

Fig. 9

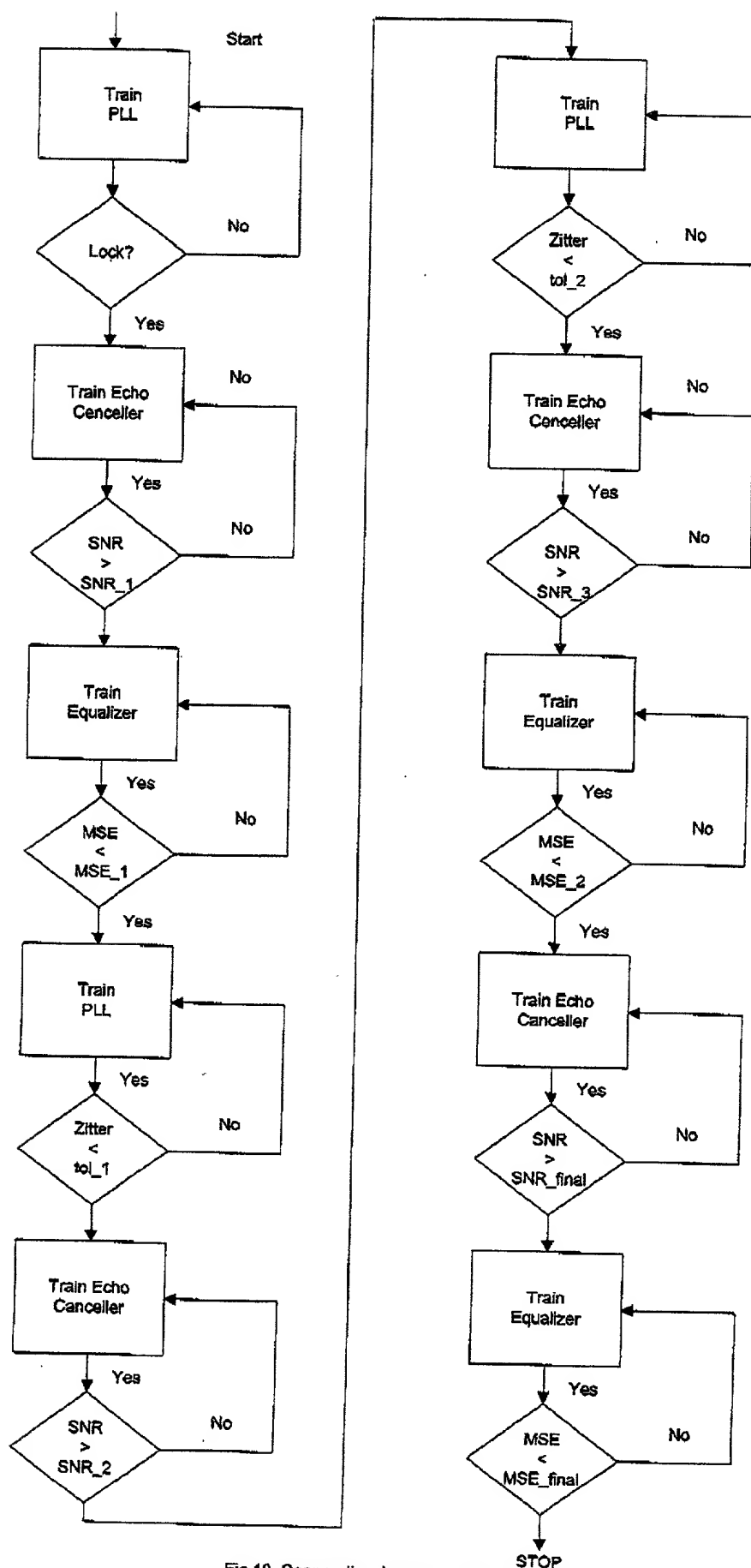


Fig 10. Cooperative, Interdependent training of various subsystems.

**COMBINED DECLARATION AND POWER OF ATTORNEY  
FOR UTILITY PATENT APPLICATION**

Attorney's Docket No.

032478-005

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I BELIEVE I AM THE ORIGINAL, FIRST AND SOLE INVENTOR (if only one name is listed below) OR AN ORIGINAL, FIRST AND JOINT INVENTOR (if more than one name is listed below) OF THE SUBJECT MATTER WHICH IS CLAIMED AND FOR WHICH A PATENT IS SOUGHT ON THE INVENTION ENTITLED:

RELAXED, MORE OPTIMUM TRAINING FOR MODEMS AND THE LIKE

the specification of which

(check one)



is attached hereto;



was filed on \_\_\_\_\_ as

Application No. \_\_\_\_\_

and was amended on \_\_\_\_\_ ;  
(if applicable)

I HAVE REVIEWED AND UNDERSTAND THE CONTENTS OF THE ABOVE-IDENTIFIED SPECIFICATION, INCLUDING THE CLAIMS, AS AMENDED BY ANY AMENDMENT REFERRED TO ABOVE;

I ACKNOWLEDGE THE DUTY TO DISCLOSE TO THE OFFICE ALL INFORMATION KNOWN TO ME TO BE MATERIAL TO PATENTABILITY AS DEFINED IN TITLE 37, CODE OF FEDERAL REGULATIONS, Sec. 1.56 (as amended effective March 16, 1992);

I do not know and do not believe the said invention was ever known or used in the United States of America before my or our invention thereof, or patented or described in any printed publication in any country before my or our invention thereof or more than one year prior to said application; that said invention was not in public use or on sale in the United States of America more than one year prior to said application; that said invention has not been patented or made the subject of an inventor's certificate issued before the date of said application in any country foreign to the United States of America on any application filed by me or my legal representatives or assigns more than twelve months prior to said application;

I hereby claim foreign priority benefits under Title 35, United States Code Sec. 119 and/or Sec. 365 of any foreign application(s) for patent or inventor's certificate as indicated below and have also identified below any foreign application for patent or inventor's certificate on this invention having a filing date before that of the application(s) on which priority is claimed:

**COMBINED DECLARATION AND POWER OF ATTORNEY**

Attorney's Docket No.

032478-005

COUNTRY/INTERNATIONAL	APPLICATION NUMBER	DATE OF FILING (day, month, year)	PRIORITY CLAIMED
			YES_ NO_
			YES_ NO_

I hereby appoint the following attorneys and agent(s) to prosecute said application and to transact all business in the Patent and Trademark Office connected therewith and to file, prosecute and to transact all business in connection with international applications directed to said invention:

William L. Mathis	17,337	R. Danny Huntington	27,903	Gerald F. Swiss	30,113
Robert S. Swecker	19,885	Eric H. Weisblatt	30,505	Michael J. Ure	33,089
Platon N. Mandros	22,124	James W. Peterson	26,057	Charles F. Wieland III	33,096
Benton S. Duffett, Jr.	22,030	Teresa Stanek Rea	30,427	Bruce T. Wieder	33,815
Norman H. Stepno	22,716	Robert E. Krebs	25,885	Todd R. Walters	34,040
Ronald L. Grudziecki	24,970	William C. Rowland	30,888	Ronni S. Jillions	31,979
Frederick G. Michaud, Jr.	26,003	T. Gene Dillahunt	25,423	Harold R. Brown III	36,341
Alan E. Kopecki	25,813	Patrick C. Keane	32,858	Allen R. Baum	36,086
Regis E. Slutter	26,999	Bruce J. Boggs, Jr.	32,344	Steven M. du Bois	35,023
Samuel C. Miller, III	27,360	William H. Benz	25,952	Brian P. O'Shaughnessy	32,747
Robert G. Mukai	28,531	Peter K. Skiff	31,917	Kenneth B. Leffler	36,075
George A. Hovanec, Jr.	28,223	Richard J. McGrath	29,195	Fred W. Hathaway	32,236
James A. LaBarre	28,632	Matthew L. Schneider	32,814		
E. Joseph Gess	28,510	Michael G. Savage	32,596		

**21839**and: Joseph P. O'Malley, Reg. No. 36,226

Address all correspondence to:

**21839**

James W. Peterson

BURNS, DOANE, SWECKER &amp; MATHIS, L.L.P.

P.O. Box 1404

Alexandria, Virginia 22313-1404

Address all telephone calls to: James P. O'Malley at (650) 622-2300.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

FULL NAME OF SOLE OR FIRST INVENTOR	SIGNATURE	DATE
BIJIT HALDER		07/26/2000
RESIDENCE	CITIZENSHIP	
870 E. El Camino Real, Apt. 158, Mountain View, CA 94040	India	
POST OFFICE ADDRESS		
870 E. El Camino real, Apt. 158, Mountain View, CA 94040		